Printable page generated Tuesday, 6 Feb 2024, 08:41

Guide to TT284 Server Accounts

Introduction

The module team has set up a server that is running PHP and MariaDB. Every TT284 student has an account on the server that provides all the server facilities you need to complete work for Block 2, Block 3 and the EMA.

At the end of Block 2 you will be asked to complete a tutor-marked assignment (TMA) which will include uploading files to your TMA 02 folder on the TT284 server. After the cut-off for TMA 02 you will find you can no longer upload files to your TMA 02 folder (unless you have agreed an extension with your tutor). After the cut-off date for the Block 3 TMA you will also not be able to upload files to your TMA 03 folder on the TT284 server.

The EMA may also instruct you to upload certain files to the TT284 server. These must be added to your EMA folder on the TT284 server before the cut-off date for the EMA.

You can access your server account at: https://oucu.tt284.open.ac.uk where ‘oucu’ should be replaced by your Open University computer username (OUCU). Your OUCU is available from your Student Home pages by going to ‘Profile’ | ‘Update personal details’ and choosing ‘Contact details’. The Computing Helpdesk will be able to help you if you cannot find it. Note that your OUCU is *not* the same as your Personal Identifier.

The TT284 server is provided to support your study of TT284, please *do not* upload any materials that are not directly related to TT284 or which might consume unreasonable resources on the server (memory or CPU). If you are found to have done so then you may be subject to disciplinary action and/or you may receive a penalty on your marks.

If you are in any doubt about how you should use your server account then contact your tutor or post your question to the Block 2 forum or the Block 3 forum, as appropriate.

The structure of your TT284 server account

This section of the guide describes the structure of your server account and some of the features that you are likely to use in your work on this module.

When you first log in to the TT284 server you should see your welcome page (Figure 1). This page provides you with your credentials (username and password) to use with the MariaDB database. See the practical activities in Block 2 Part 5 for instructions on how to use these.

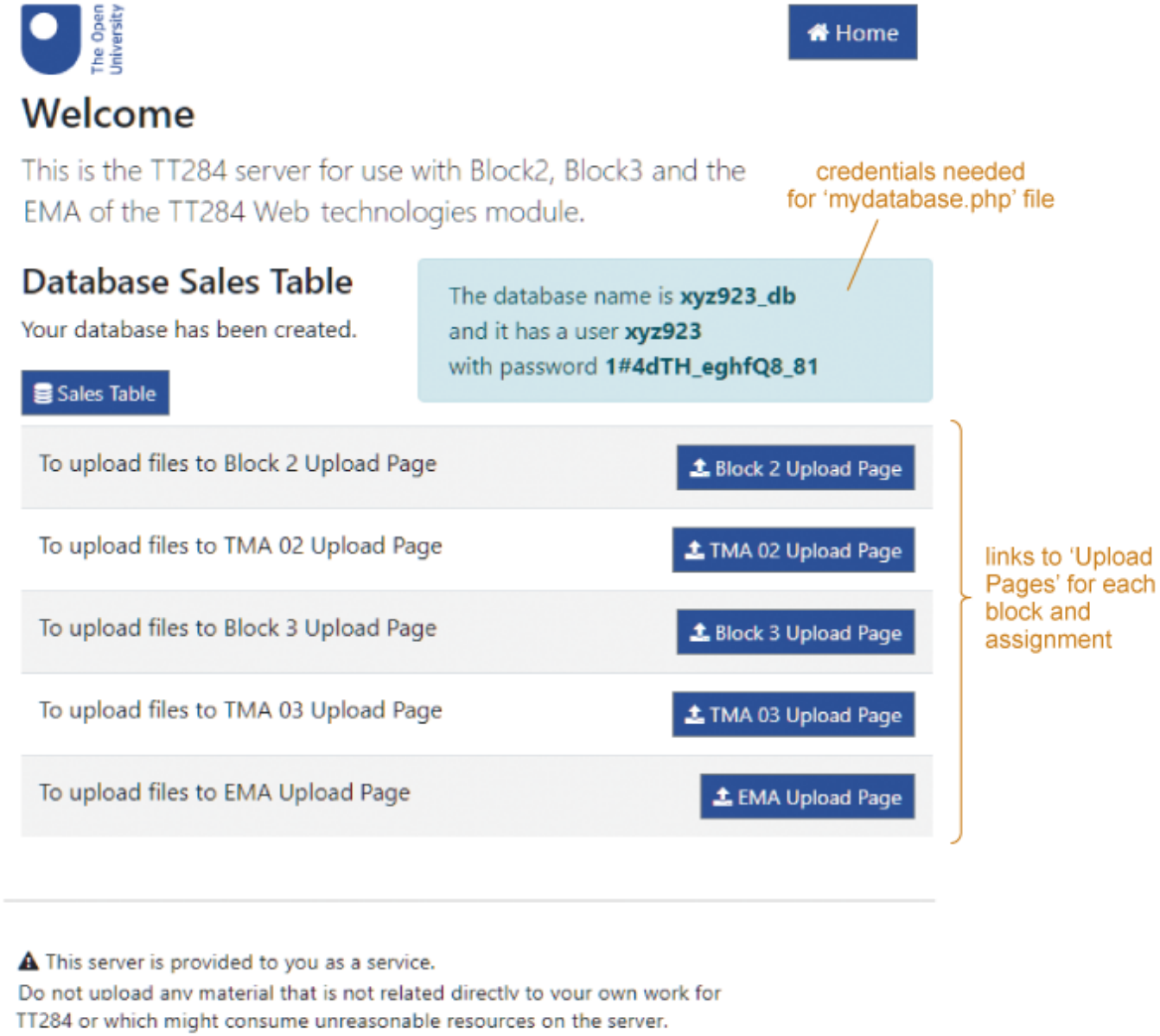

Figure 1 Example of an account home page on the tt284.open.ac.uk server

Following any of the 'Upload Page' links will lead you to a list of all the files that have been uploaded so far. The contents of this list will depend on what you have done, but a typical example is shown in Figure 2.

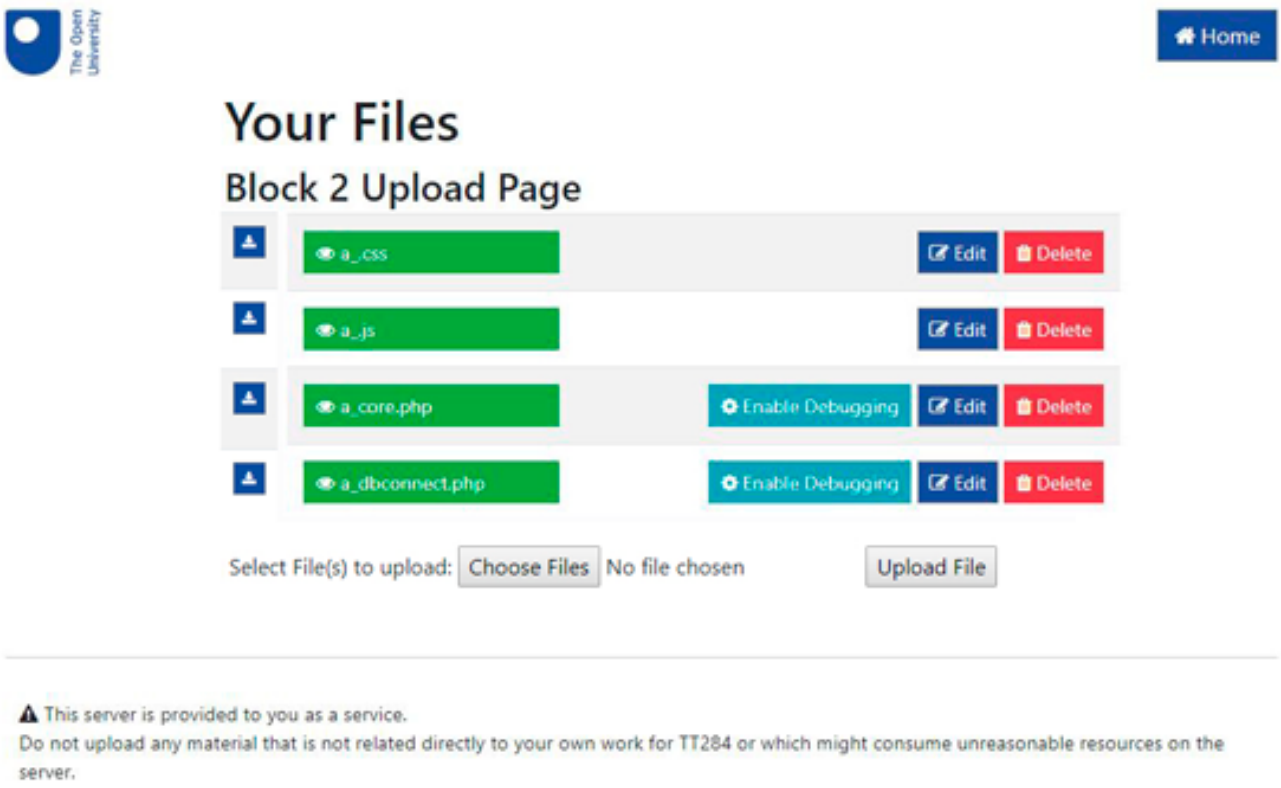

Figure 2 Example of a folder page within a student account

Maximise

Uploading files

The first thing you will want to do is to upload files. At the bottom of a given ‘Upload Page’ you will see a button called ‘Choose Files’, and clicking on this allows you to select the one(s) that you’d like to upload.

Note that you can use the normal techniques for your Operating System to select multiple files to upload. So on a PC, click the first file to upload then hold down ‘Ctrl’ and click any additional files you wish to upload at the same time (or hold down ‘Shift’ and click another file to select it plus all the other files between those two). When you have made your selection, the text to the right of the ‘Choose Files’ button will show the name of the file in question (if there is just one) or how many files have been chosen, as shown in Figure 3.

Figure 3 Example after choosing 16 files to upload

When you are happy with your selection, click ‘Upload File’. The files will be uploaded subject to two limitations: *Only the first 20 files will be uploaded*. This is to avoid accidentally uploading a large number of files by accident. Any additional files will be ignored, with no warning. To upload more than 20 files, upload them in batches. Files will be uploaded *as long as they don’t already exist on the server*. If they do, you will get a warning, as shown in Figure 4.

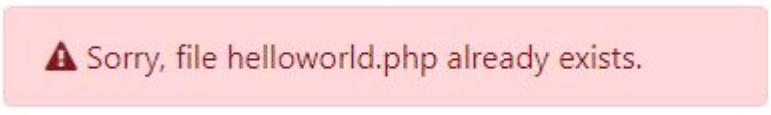

Figure 4 Warning if a file exists

This feature will help you to avoid overwriting any edits you have made on the server copy of a file. If you really want to replace the server file with the copy from your local machine, delete it from the server first.

A note on folders

Often, we place images in an images folder, to separate them from the HTML files. Sometimes CSS is also placed in a folder.

The TT284 server space does not support folders and all of your files must be contained in the upload space. As our projects are small that should not be a problem.

Downloading files

The next few screens of this guide discuss features that are available from the ‘Upload Page’ for a given aspect of the module (as was shown in Figures 1 and 2).

Figure 5 shows an example of the ‘Upload Page’ for Block 2. The column on the left contains links to allow you to download the files. Those are particularly important for PHP files where any request to the server would normally return only the results of executing the PHP code, not the PHP code itself.

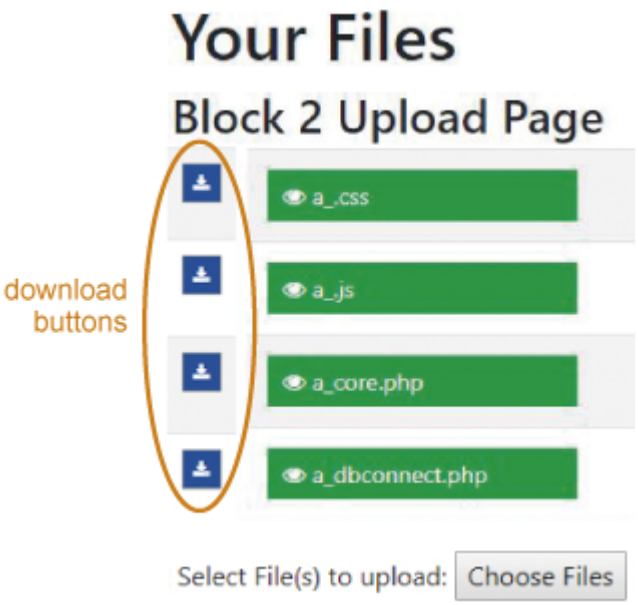

Figure 5 The download buttons on an example ‘Upload Page’

Viewing files

The next column lists the file names. Clicking here will request the webserver to serve the file. Those ending ‘.css’, ‘.js’, ‘.html’ or ‘.htm’ will simply be delivered to the browser as plain text. The ‘.css’ and ‘.js’ files are not processed by the browser and display as their plain text content. The ‘.html’ and ‘.htm’ files should contain HTML and that will be rendered by the browser as a web page.

Those ending '.php' are processed by the server. The PHP elements are executed and the resulting text delivered to the browser. That text is generally HTML and so is then rendered by the browser as a web page.

Debugging

When the PHP is processed, there may be errors. It is considered bad practice to display those errors to the user, as a hacker may use the information to gain insight into how to attack the system. However, that makes finding errors (debugging) much harder.

To facilitate debugging in your server account you can click the 'Enable Debugging' button that appears next to some files that you have uploaded. (It is only available for PHP files.) When clicked, it adds a directive to the start of the PHP file to tell the server that in this case errors should be displayed. That line of code looks like this:

```
<?php error_reporting(E_ALL); ini_set('display_errors', 1); // DO NOT MODIFY THIS LINE ?>
```

Now, when you click the button to request the file (i,e, the green button with an eye icon followed by the filename in question) the HTML output will include error reporting of most problems. The most serious syntax errors that prevent PHP from executing the code at all will still result in a blank page. In that case editing the file on the server (see next section) may help you to identify and fix the problem.

You can turn the debugging feature off by clicking the button again (which will be now called 'Disable Debugging' if you have previously enabled it for a given file).

Editing files

The 'Edit' button allows you to edit a file that you have uploaded to the server (Figure 6). This is a very useful feature, but do be clear that it is *only the server copy that is changed*. That can be useful – you can try things out and if you make mistakes, just delete the server copy and upload a fresh copy from your local machine. However, if you make edits you wish to save back to your person computer, you will need to download the server copy to your local machine.

Figure 6 Example of a '.php' file open in the editor

Maximise

Features of the server editor tool

The editor is easy to use, but also has some useful features to make it easy to edit to the files while they are on the server.

Line numbers

Error reports will often indicate that an issue occurred on a specific line number in your code. These numbers are shown in the grey column on the left in the server editor, and will help you to find the offending line quickly.

Syntax highlighting

The editor has some understanding of the syntax of HTML, CSS, JavaScript and PHP, so can colour the different elements to make the code easier to read.

Code folding

Notice the small triangles in lines 2,3 and 7 in Figure 6. Clicking on those collapses the tag with its partner, hiding the contained text.

This feature can be used, for example, to hide the <head> while you are working on the <body>, and therefore allow you to focus on smaller sections of the code at any one time. Similarly, elements of a CSS stylesheet, PHP or JavaScript can be folded.

Auto completion

If you enter an HTML tag, its closing partner will be automatically added. Parenthesis and braces will have closing partners added.

Error and warning flags

As shown in Figure 7, these are displayed to the left of the line numbers. A yellow triangle containing an exclamation is a warning, while a red square containing a cross is an error. Hover over these to see details of the issue.

Figure 7 Example of a warning and error in the editor

If a PHP file always generates a blank page when executed, look for syntax errors in the editor as indicated in Figure 7. Note that some warnings are about lack of feature support in old browsers, or offer opinionated recommendations, and can be disregarded.

Although the editor has some understanding of syntax, it is not a replacement for validating HTML or CSS files.

Save feature

Finally, at the top right there is a save button which will save the edited text back to the server. The cross next to it would discard the file without saving the changes.

Deleting files

The ‘Delete’ buttons next to files in the list on your upload pages remove the file from the server. Be clear that although it asks for confirmation, if you go ahead with the deletion then the file will be removed at once, and there is no means to recover it.

Of course, a copy should be on your local machine and it is only the server copy which is deleted, so you can just upload a fresh copy if needs be. However, any edits that you have made to the file directly on the server since the last download will be lost.

Workflow

It is entirely up to you how you work, but it is generally best to avoid having to upload your file excessively.

Traditionally we edited files on a local machine, then uploaded them to the server to test. We then corrected issues or added functionality locally, and uploaded the file again. That repeated uploading is very tedious, and by using the editor on the server you can change and test your files much more quickly. That approach also supports a workflow with small, incremental changes to code that are being tested regularly, and we recommend that.

Here are a few tips for effective working.

- Start by either downloading the example file (if one is provided) to your local machine or creating a file on your local machine. If it is a new file, put a small part of the code in – as much as you feel comfortable writing without testing.
- Upload that file to the server and work to develop it there.
- Develop code incrementally (add a small amount of code to it, typically to achieve one task).
- Test your code regularly, and resolve issues before progressing.
- Download a copy to your local machine at each significant point in development, just in case you need to return to a given version at any stage in the development.
- With HTML/CSS pages, copy the whole text into the validator direct input windows and validate regularly.
- Sometimes, in trying to fix errors, we lose track and everything seems to become a problem. If that happens, delete the server file and upload the most recent download from your local machine to return to the point where you knew what was happening.

However, you may choose to do more offline and if you are comfortable with that, then that is fine.